

Agile Release Planning & Integrated Scheduling

Robin Pulverenti

Joint IT and Software Cost Forum

September 2021



ADEPT
FORCE GROUP
STRATEGY TO SOLUTION

Poll 1 – Program Stage

- What lifecycle stage is your program?
 - Planning (Requirements Definition, Pre-Selection)
 - Execution (New Development or Modernization Acquisition Program)
 - Sustainment with Enhancements
 - Sunsetting

Multiple Selections are allowed

Presentation Assumptions

- Focus is on NEW Development Program (not existing/sustainment)
- Scaled Agile Framework (SAFe) is the base methodology with some tailoring bringing in concepts from other methodologies
- Jira is the Agile Tracking Tool for metric terms
- Release = Deliverable to the Field/Operational Environment Deployment
- Increment = Integration Test Point after multiple Sprints as Epics complete

Software Development Paradigm

Agile Manifesto values “Individuals & Interactions over Processes & Tools”



Agile

- Small Scale
- Single Development Team
- Volatile Requirements
- Software-Only Environment
- Iterative, Incremental Design & Development



- Large Scale
- Risk-Averse
- Mission-Critical System
- Strong Management Oversight & Hierarchical Governance
- Top-Down Design

CMMI

Capability Maturity Model Integration



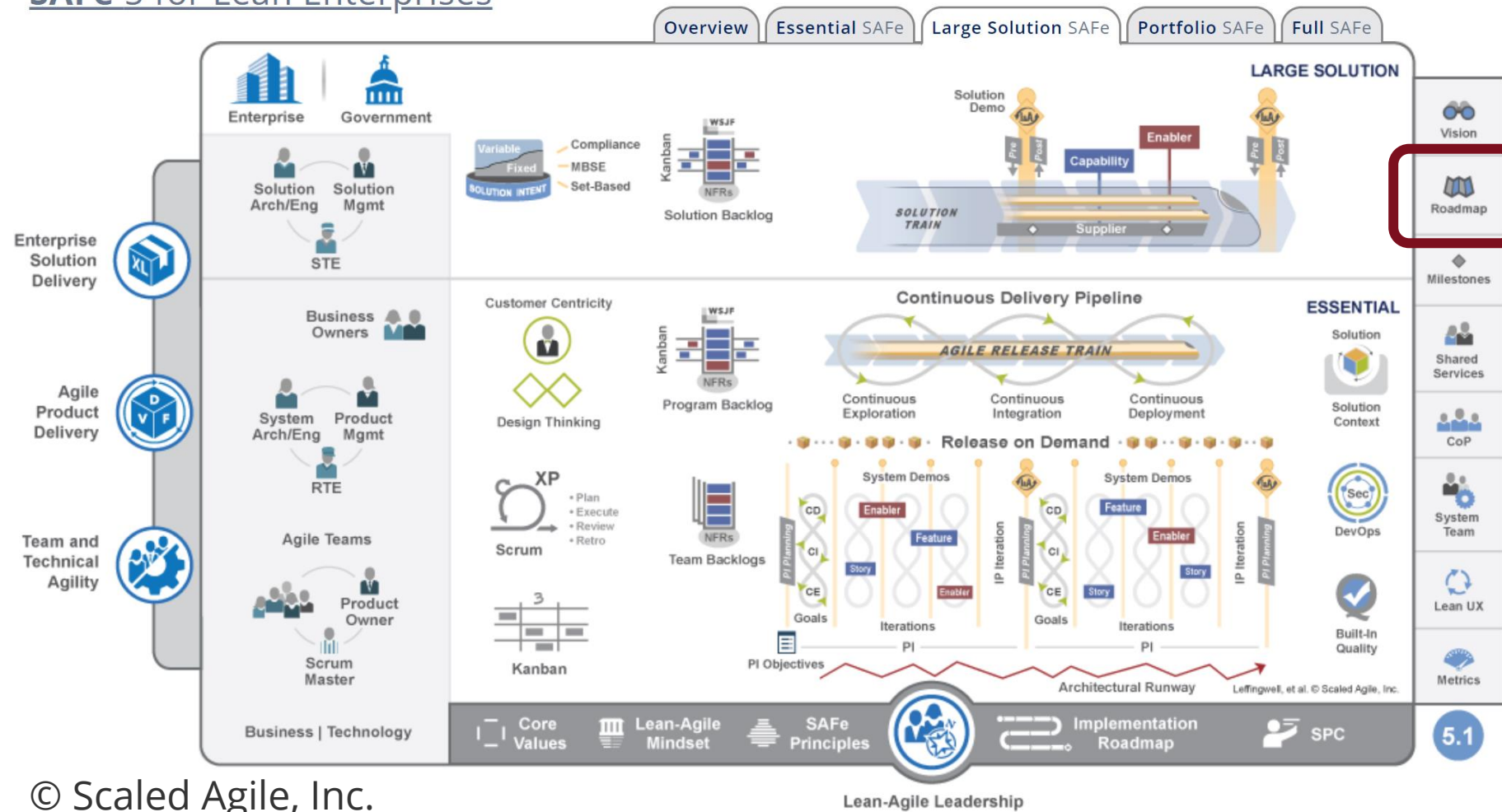
However, there needs to be a balance to enable a reliable CI/CD SecDevOps pipeline

Release Planning Considerations

- Product Roadmap & Release Planning
- Capability & Epic Definition (by Deliverable)
- Integrated Schedule
- Supporting Performance Metrics
- Forecasting Considerations

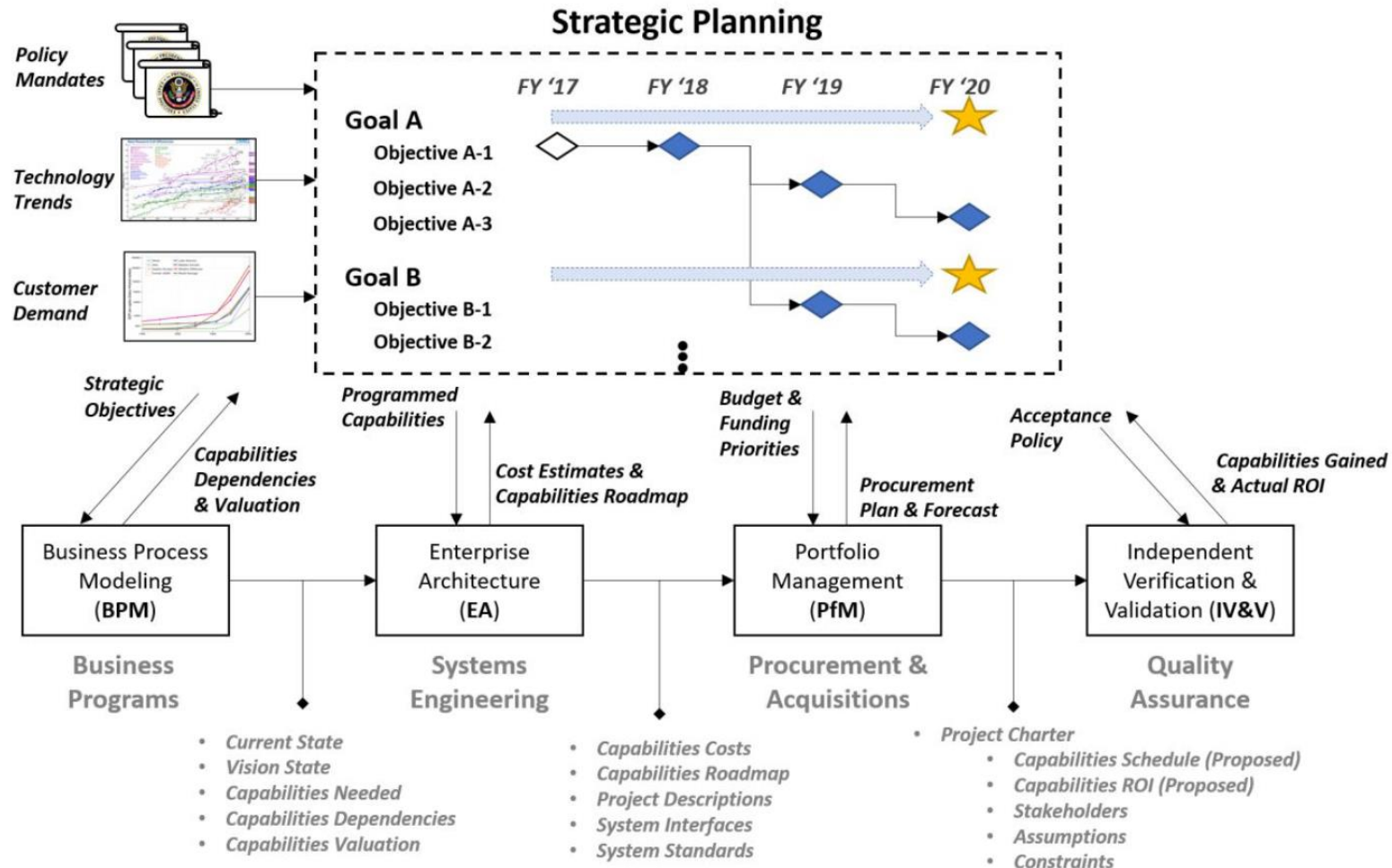
The Roadmap is part of SAFe

SAFe 5 for Lean Enterprises



- Key element of Product Development
- Schedule of Events that communicates planned Solution Deliverables
- Often neglected part of the Agile Training because it is part of the systems definition process

Portfolio Roadmap



Solution Roadmap

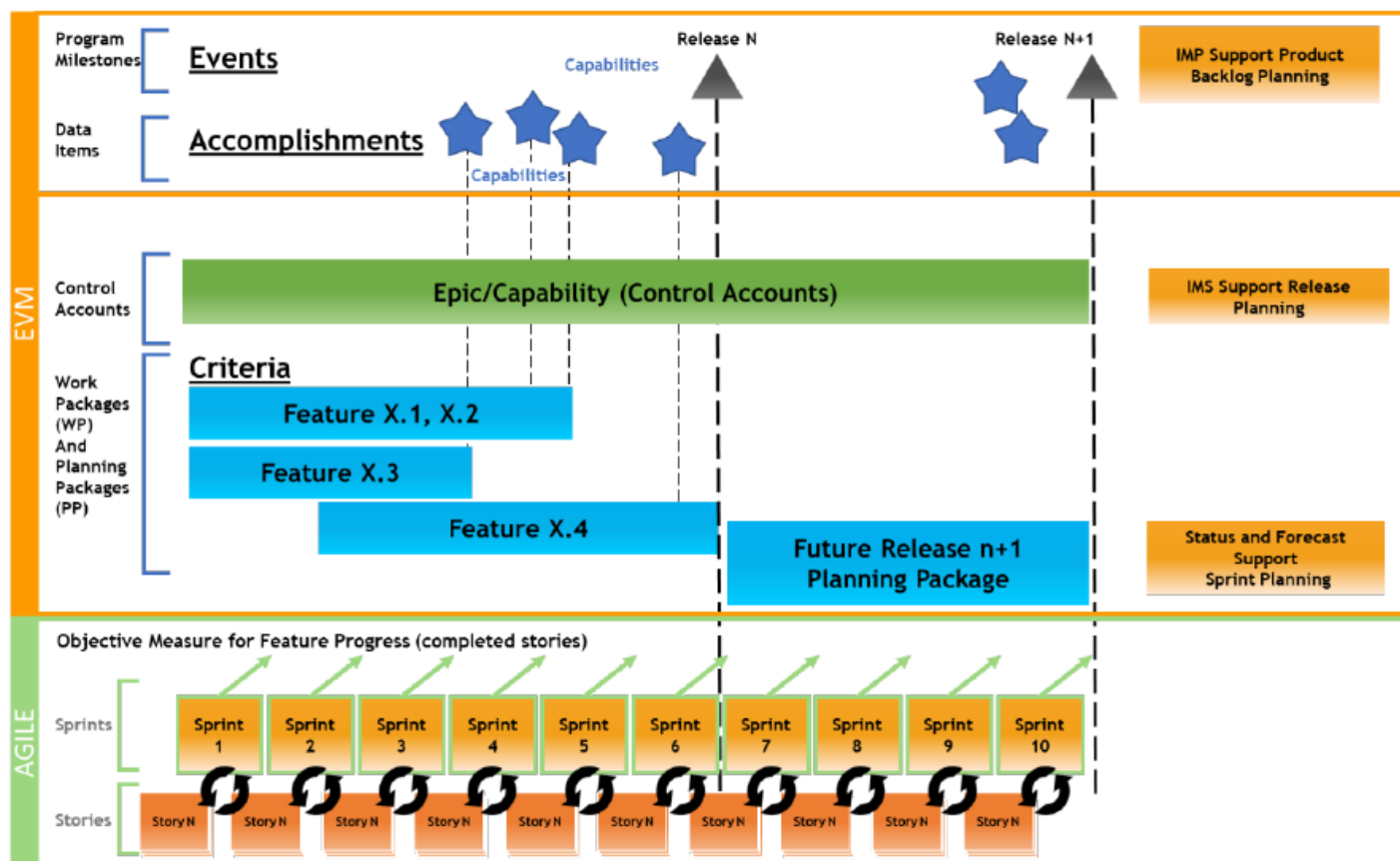


Figure 2-1: Agile IMP Event to EVMS Hierarchies. In this example, IMP events are equivalent to Customer Releases, with Significant Accomplishments and Accomplishment Criteria representing delivered capabilities delivered in Work Packages where Features are implemented.

Graphic From: NDIA An Industry Practice Guide for Agile on Earned Value Management Programs v1.3, 26-MAY-2019

Poll 2 – Product Decomposition

- Do your Agile programs use a “Prescribed” WBS (MIL-STD-881) or is it a “Contractor Defined” WBS?
 - Prescribed
 - Contractor Defined
 - We don’t require a WBS on our Agile Projects

Capability (Work) Breakdown Structure & Requirements Traceability

Program Vision, Mission, ConOps

- Global/System Level Requirements
- SAFe Portfolio Roadmap, Program Strategic Plan

Deliverable (Fielded Release)

- Deliverable Scope Statement & Success Criteria
- SAFe Solution Roadmap

System-Level Capabilities

- Logical Grouping of Business Processes
- SAFe PI Roadmap

Business Process (Operational) Capabilities

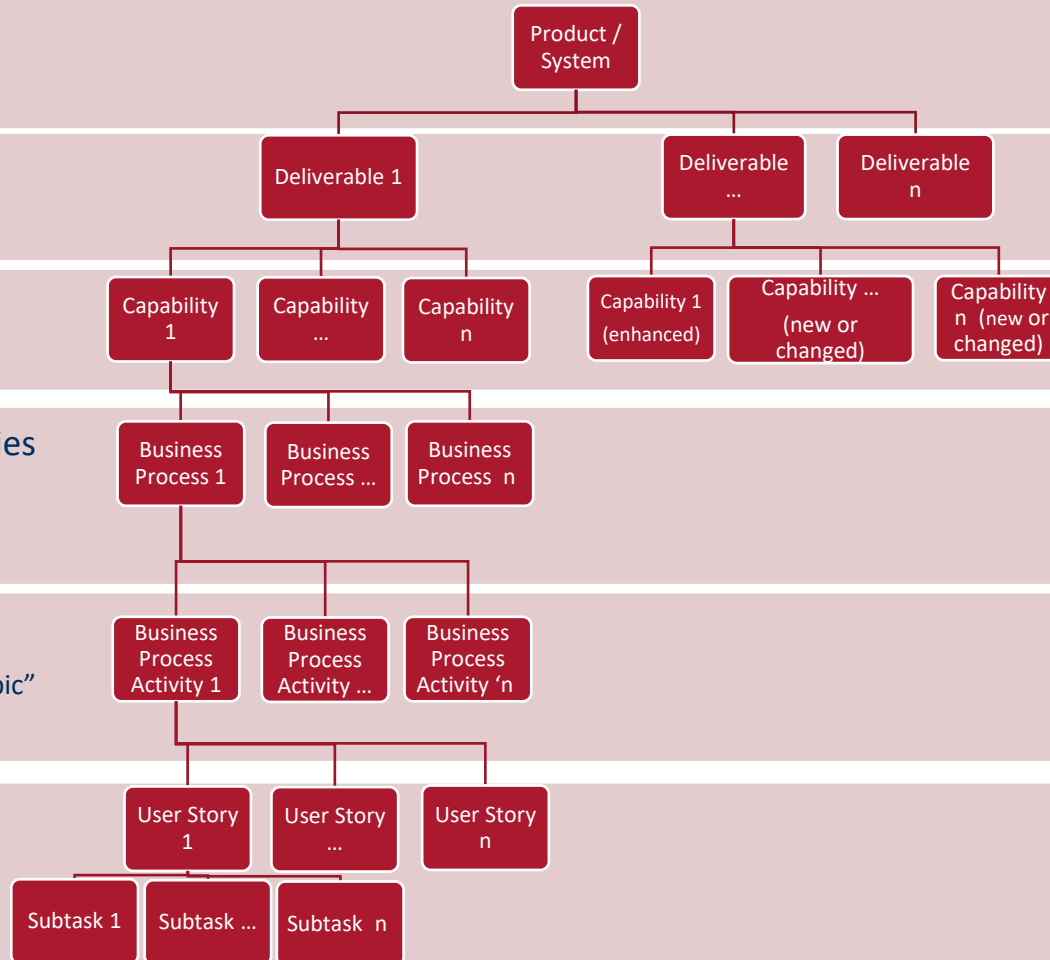
- Requirements identified apply to all Activities
- "Control Account" for EVM, Agile/Jira "Initiative"
- Summary Task in the Integrated Schedule

Business Process (Operational) Activities

- Requirements identified apply to all Stories
- "Work Package" for EVM, Agile "Feature" / Jira "Epic"
- Task in Integrated Schedule

User Story / Subtask

- Requirements apply are specific to use case
- Tracked in Agile Tool
- Performance aggregated to Epic

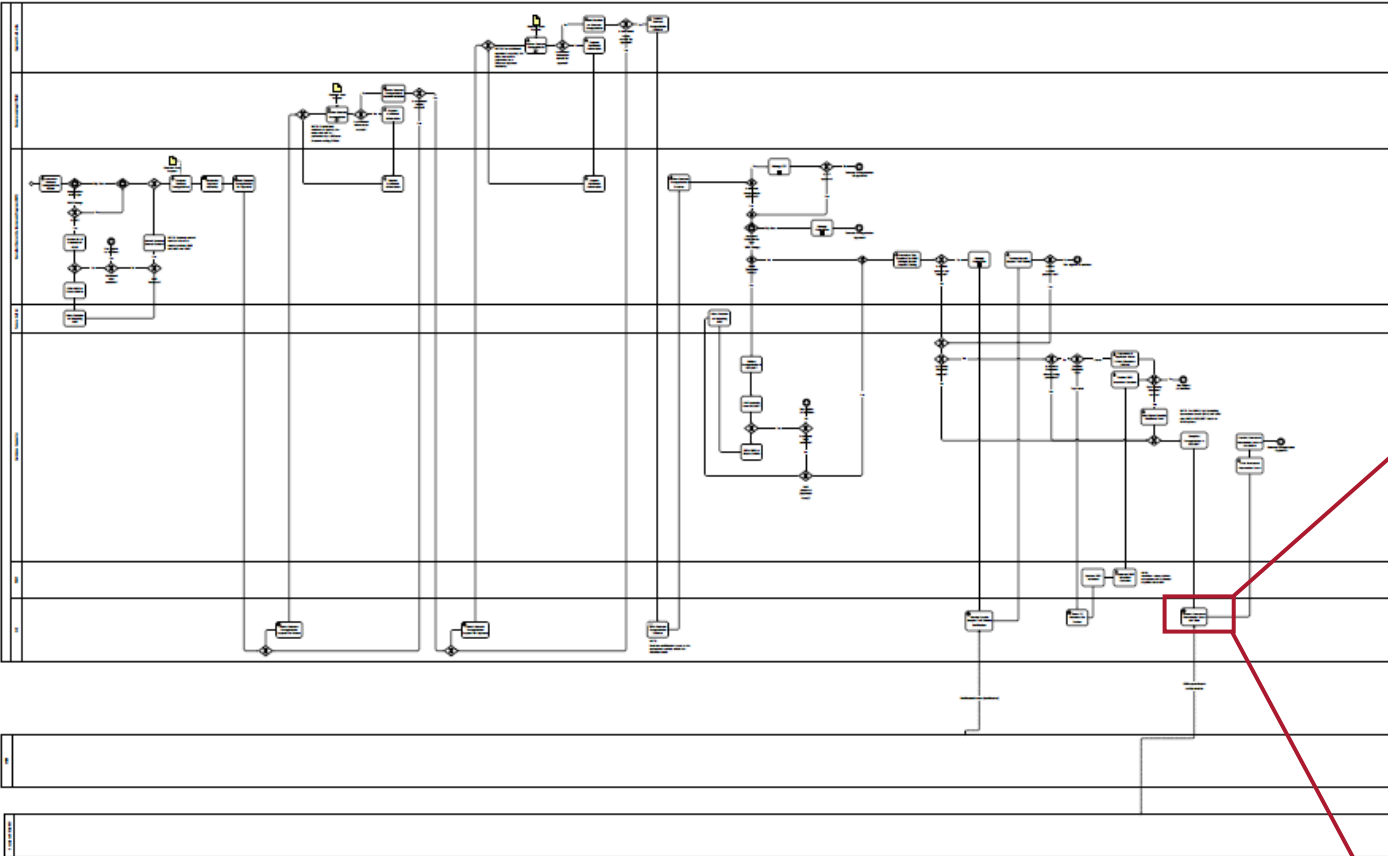


Requirements should be decomposed aligning to:

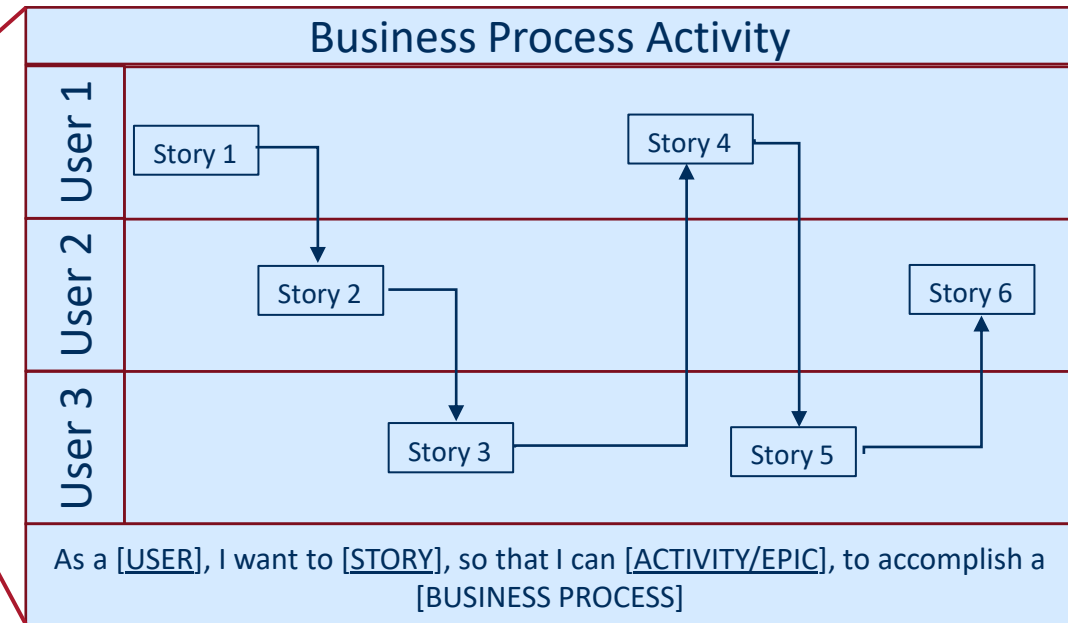
- 1) Phased implementation by Deliverable
- 2) To the highest level where it applies in the capability breakdown (all children inherit the parent requirements)

Detailed Architecture Emerges from an Intentional Design

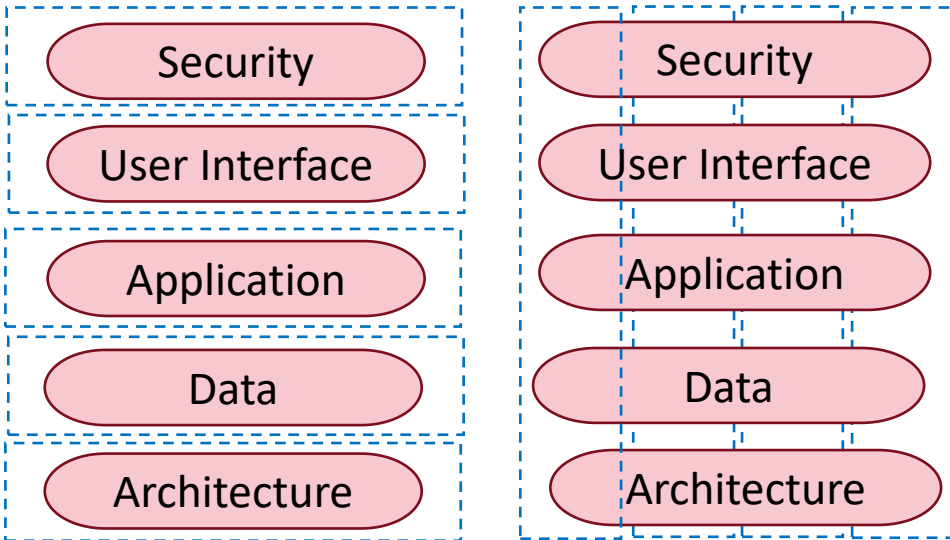
Business Process



- Intentional Design: Operational Capabilities and Activities are defined with acceptance criteria for a Deliverable with associated Inputs/Outputs
- Emergent Design: User Stories (and subtasks) are identified through progressive elaboration to meet the needs of the Deliverable



Business Process Descriptions



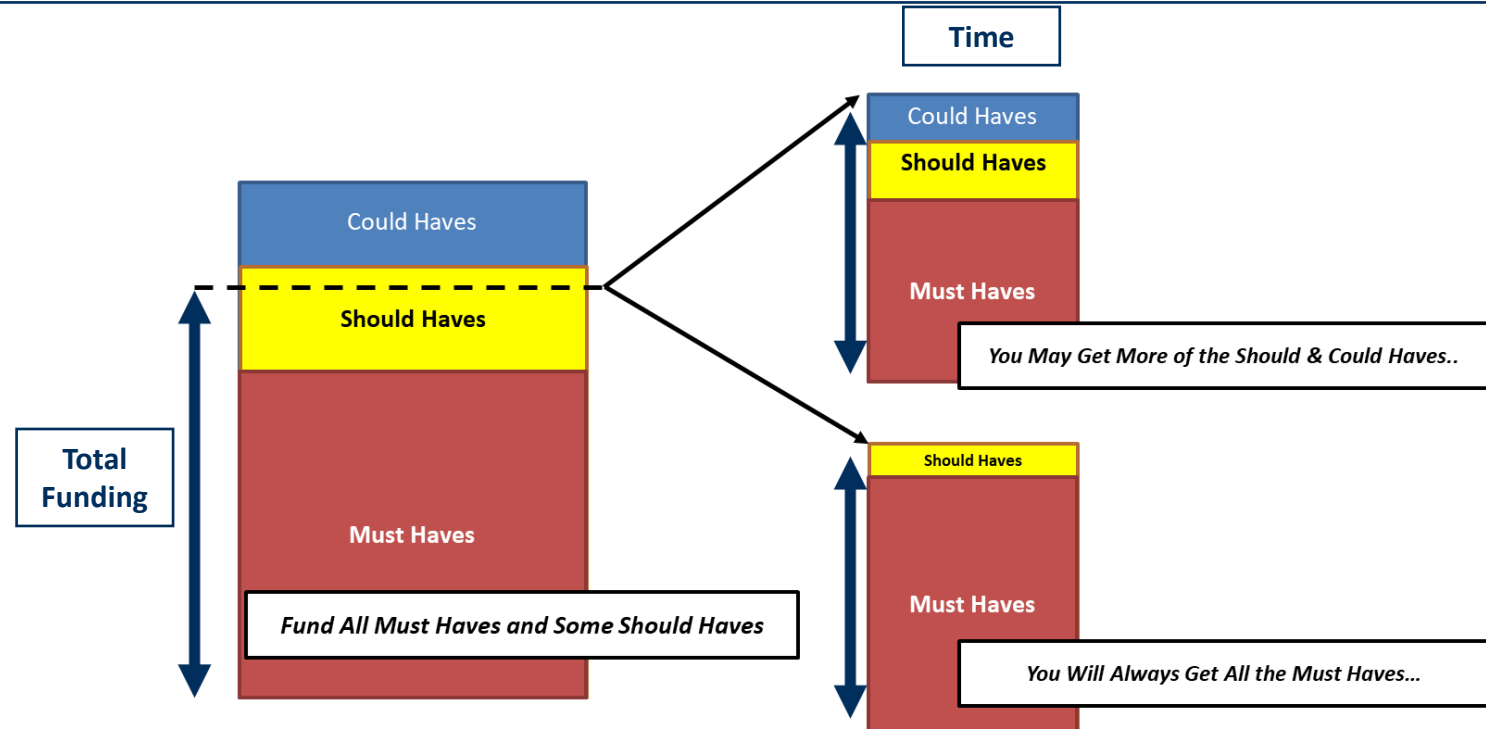
Work Breakdown Structure
(How it's typically built)

Capability Based
How the user consumes it

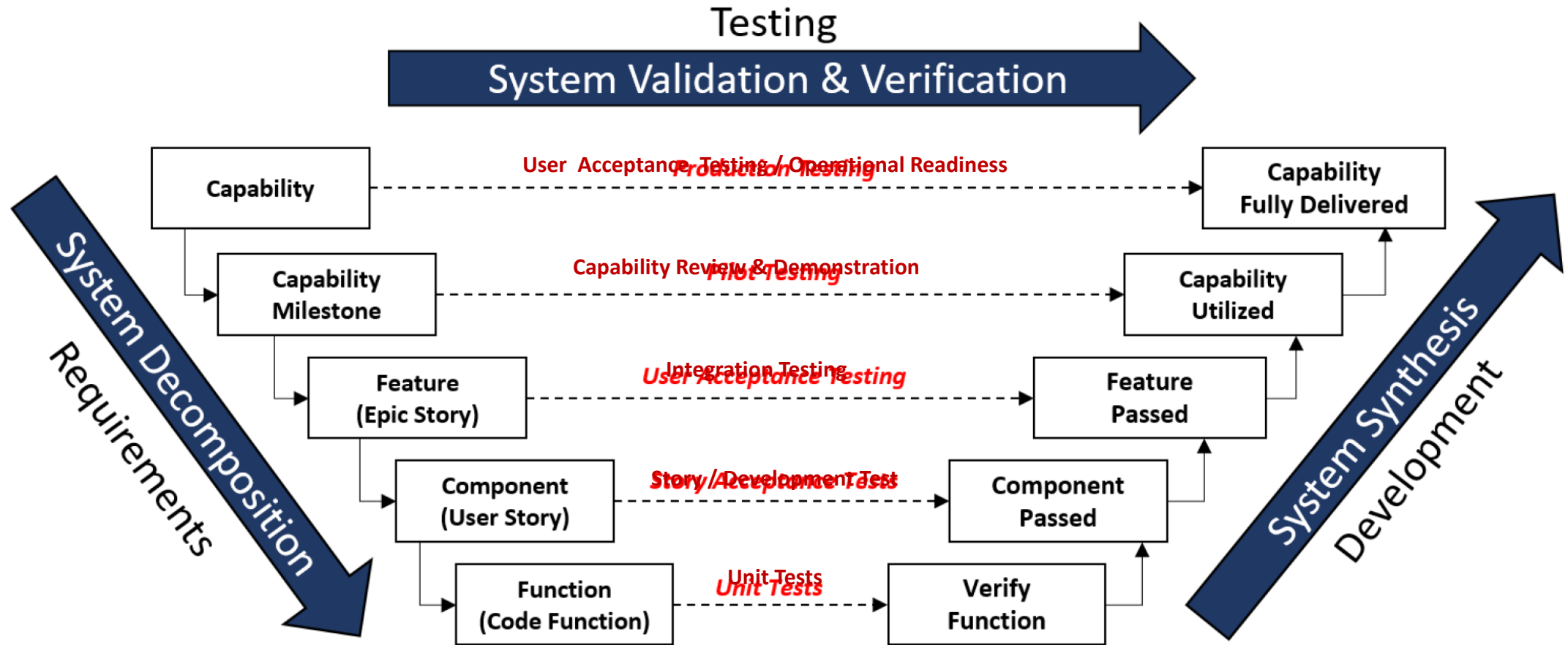
Business Process Descriptions

- Narrative of functionality and limitations by capability from the User perspective
- Success criteria defined for each integration point (increment, release)
- “[At this point in time], the system/user will be able to with the following limitations or constraints.”

Scope Buffer Reduces Estimate Padding On Individual Epics or Stories, while guaranteeing the delivery of MVP Features by providing trade space if needed as the detailed design & discovery continues through the sprints or technical debt builds and needs to be addressed



The "Agile V"



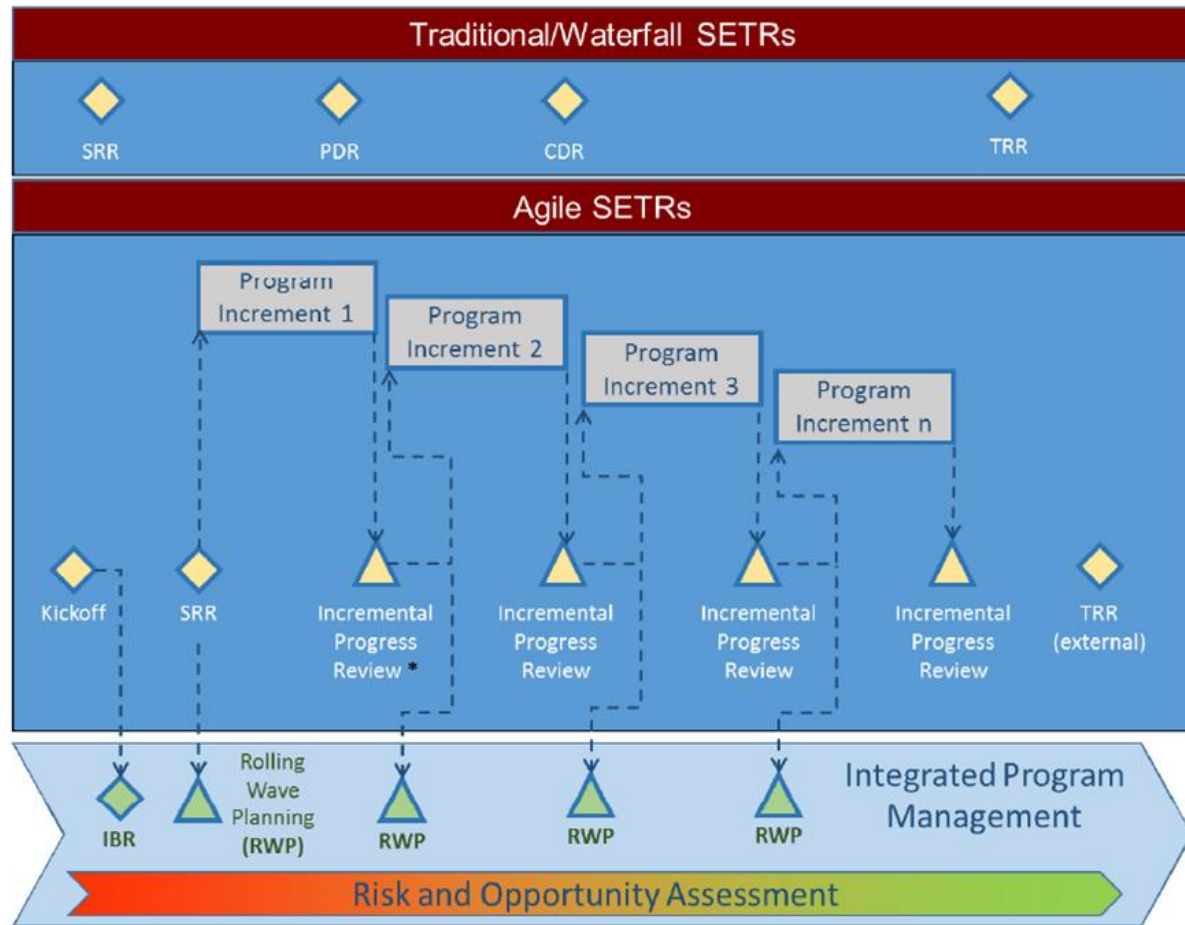
Poll 3 – Integrated Schedule

- Does your program require an Integrated Master Schedule compliant with Government (GAO, DCMA) Best Practices?
 - Yes – compliance with critical path
 - Sort Of – compliance and critical path are not enforced
 - No – we are full Agile and manage with other tools

Integrated Schedule Key Points

- Development/Modernization Programs are different than Sustainment
- Sprint Calendar ≠ Critical Path
 - Use the Sprint Calendar as a Time-Phasing Guide for when to start Epics based on resource availability or backlog prioritization
 - Development will take as many sprints needed to meet the success criteria for a “minimally viable product”
- Identify Technical Dependencies Between Epics
 - Maintain Priorities Between Scrum Teams
 - Capability MVPs by Integration Point
- Identify Key Milestones & Tasks that support them
 - Technical Reviews & Documentation versions
 - Acquisition Milestones & External Reviews/Briefings
 - Coordination with External Stakeholders (Interface Partners, Cyber, FISCAM, etc)
 - Capability Integration Points & Capability Demonstrations

Program (System Engineering) Technical Reviews



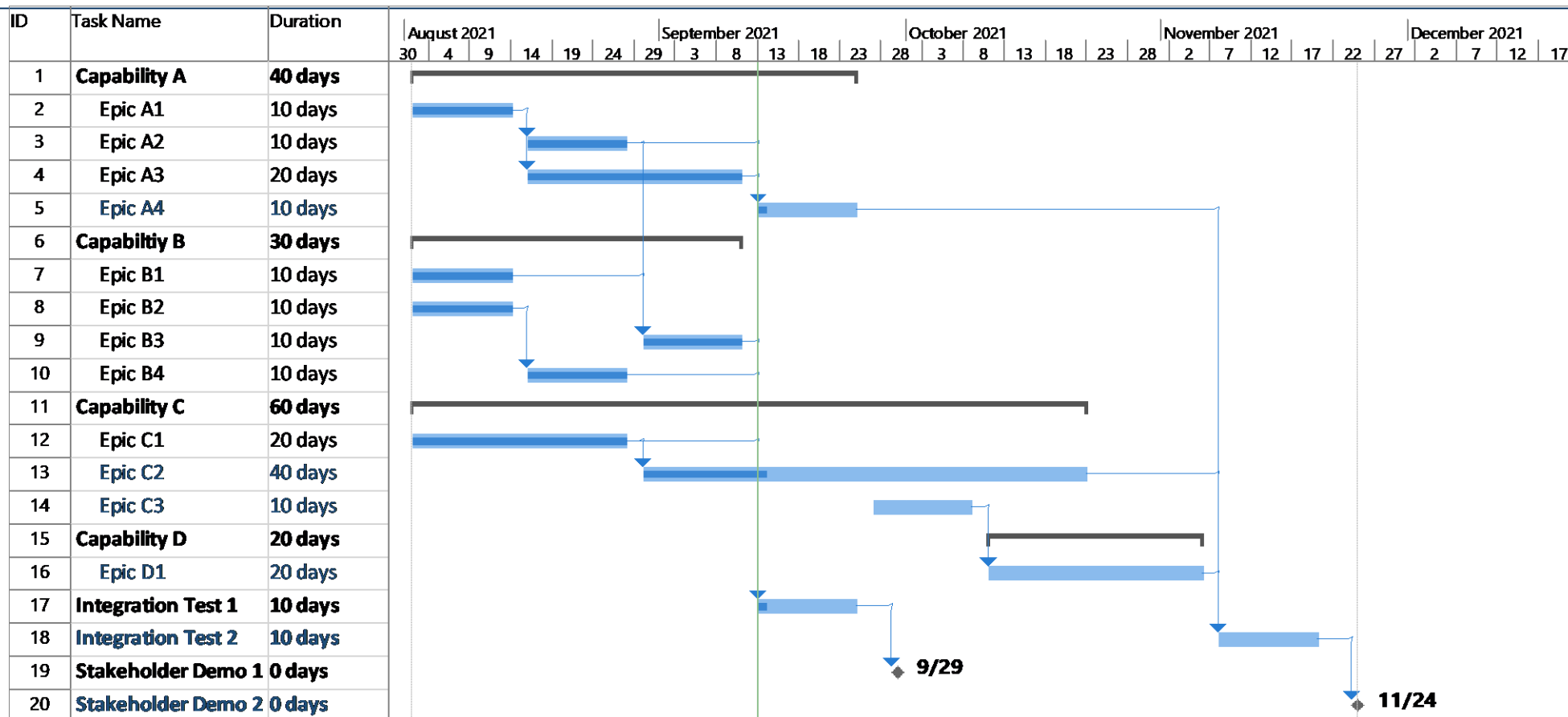
- “Big-Bang” Events (PDR, CDR) replaced (not removed) with Capability-Based Reviews and Demonstrations at appropriate Integration Points
- Integrated Baseline Review (IBR) to define Technical & Functional Baselines for the Release Deliverable, and targets for each Increment to be updated through a Rolling-Wave Plan (RWP)
- Test Readiness Review (TRR) conducted for Government/External User Acceptance Test

Graphic From: NDIA An Industry Practice Guide for Agile on Earned Value Management Programs v1.3, 26-MAY-2019

Product Documentation

- Agile Manifesto prioritizes “Working Software over Comprehensive Documentation” ... but it doesn’t replace/remove the need for documentation
- In Large-Scale Government Systems, the system documentation is part of the requirements ... and therefore, part of the system delivery
 - Documentation can be tailored to programmatic needs
 - Required Documentation can fit into and support the Agile framework
 - Documentation content evolves over time to remain in alignment with capability development
- Limiting design documentation to the “User Stories” is problematic
 - Cohesive & Holistic Documentation (System/Sub-System Views)
 - The ability to aggregate User Stories with all of the other linked issues (Defects, Test Cases, Tasks, Subtasks) can be difficult and confusing at scale
 - Low-level implementation details are great, but User Stories don’t always meet the intent & needs captured in higher-level CDRL documents
 - History & Accessibility Limitations
 - Unless the complete backlog is in a GOV tool, the history may be lost or incomplete when development ends or the Solution Provider changes
 - Not every stakeholder has access to the backlog tool
- Some Agile Development Teams might struggle with creating documentation – especially if they are used to receiving detailed design documentation post-CDR before starting work

Integrated Schedule Simplified Example



- Epics and the planned Integration Points are depicted in an Integrated Schedule
- Dependencies between Epics (as well as Float) are visible across teams to help maintain priorities
- Agile Tools are typically limited in their ability to show precedence logic and critical path

Poll 4 – Metrics Analysis

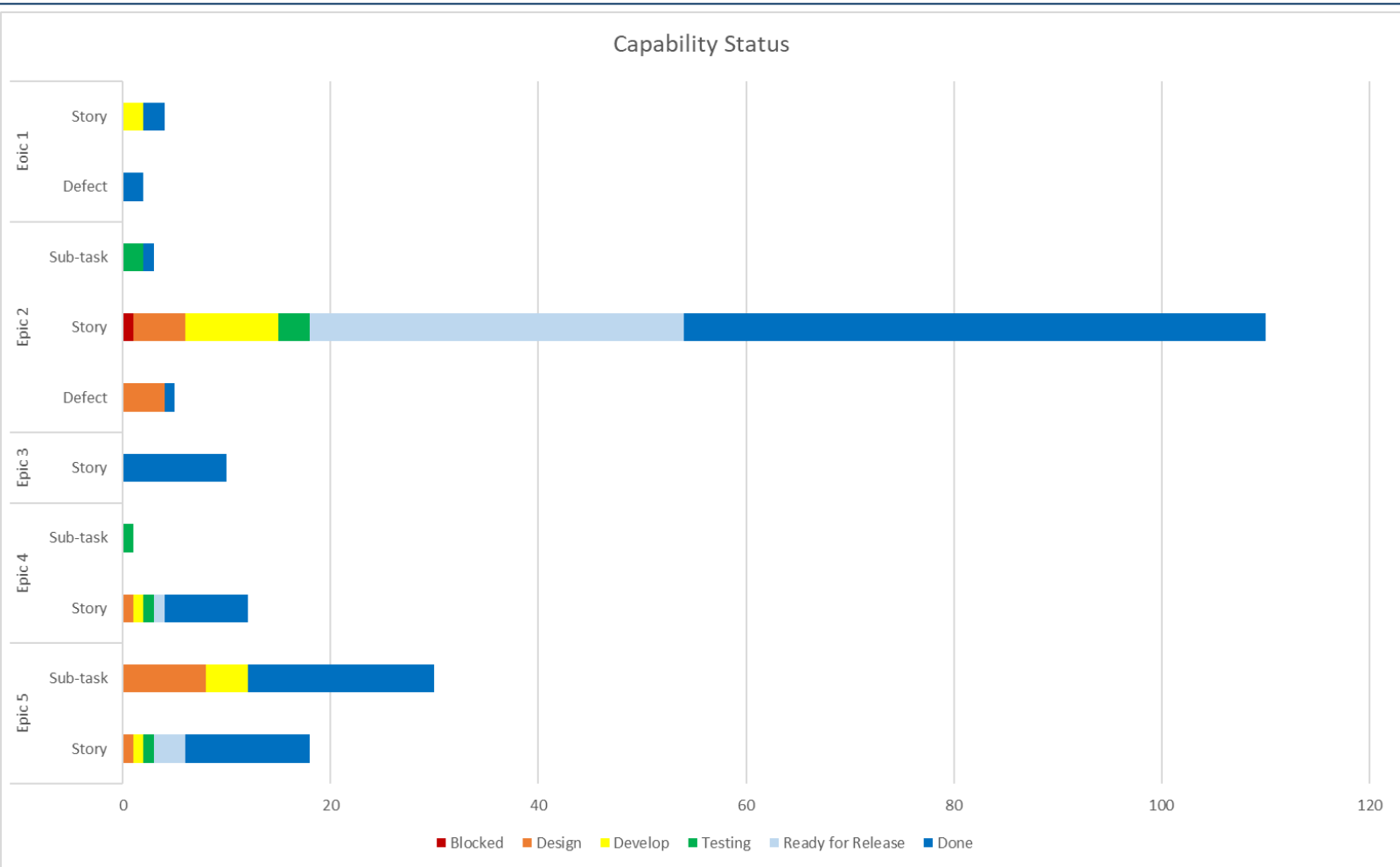
- What level of metrics tracking and analysis does your program office perform?
 - Daily Tracking with Weekly Analysis
 - Weekly Tracking and Analysis
 - We don't track metrics directly, we just review the status reports from the Scrum Masters/Team Leads
 - Other

Agile Metrics Considerations

CONSISTENCY is the Key to a High-Quality, Proactive Metrics System

- Core Metrics need to be established for the overall program
- Different Teams will have different approaches to determining Story Points and how they define their work (User Stories, Tasks, Subtasks) – acceptable as long as they are mapped into the overall program metrics
- Changes are infrequent & applied against the remaining backlog for future tracking
- Jira Data is REAL-TIME ... if you miss a download, it's gone
 - Try to download relevant filters at the same time everyday/week to avoid missed or skewed data

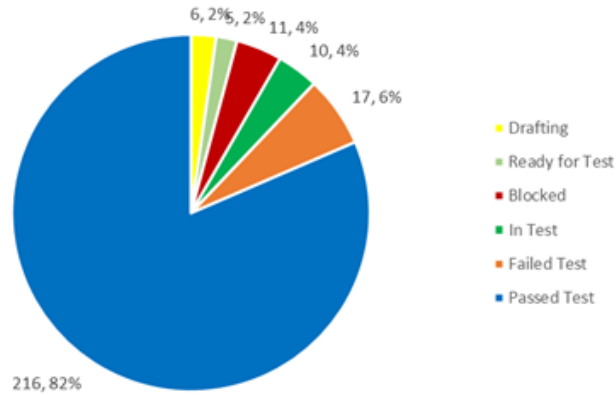
Capability / Epic Status Charts



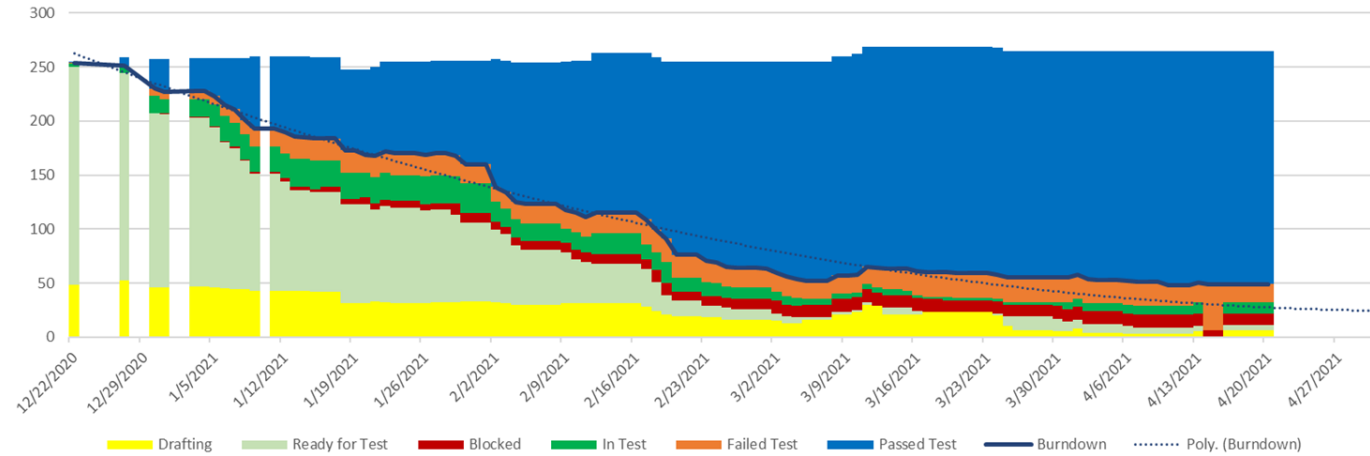
- Horizontal Stacked Bar Charts can be used to show the status of Epics that comprise Capabilities at a given point in time.
- In this example, more complex Stories that couldn't be broken down into smaller stories have sub-tasks assigned.
- Development Defects are also included in the chart as they need to close prior to migration to Integration Testing.
- Percent Complete is derived based upon agreed criteria
 - Story Points
 - Count of Issues
 - Kanban Weighting

Burn-Down & Burn-Up Charts

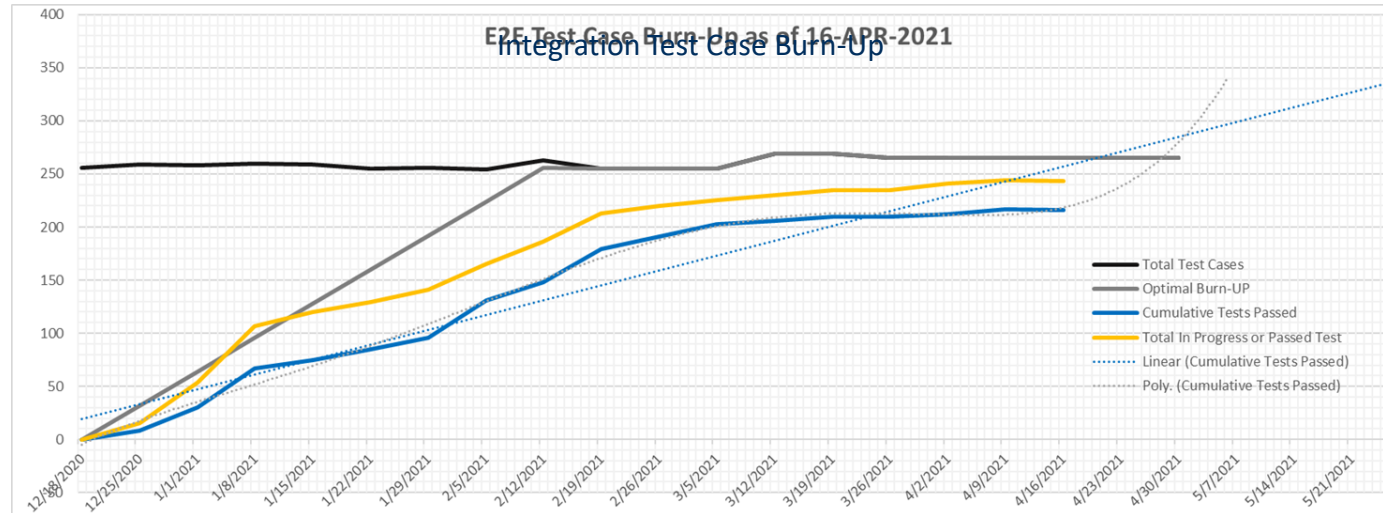
Integration Test Case Status - 20-APR-2021



Integration Test Case Cumulative Status



E2E Test Case Burn-Up as of 16-APR-2021



- Pie Charts show the status at a given point in time for work in the defined states.
- Line Charts & Cumulative Flow Diagrams (CFD) show progress made over time as work moves through the defined states established for the task type.
 - CFDs are stacked bar charts and can show more data elements cleanly (each bar is the pie chart for that date)
- Burn-Down Chart – Goal is to get to 0 (good when you have a fixed date and need to calculate needed velocity)
- Burn-Up Chart – Goal is to hit the maximum (good when you have a changing backlog)

Forecasting Considerations

- Technical Dependencies between Epics (especially when worked by separate teams) are potential risks
 - Predecessor tasks need to remain a top-priority in Sprint/Increment Planning
 - Progress of predecessor will directly impact the successor, possibly the Integration Testing Event or Critical Path
- Metrics & Metric Charts should be used to verify status, show emerging issues with velocity, & identify the need to add Sprints to complete capabilities
 - Metrics are a feedback mechanism to ensure the development is on track or requires adjustment and possible intervention
 - Inconsistency in Metric Calculations will lead to unreliable Forecasts

Project & Agile Points of Failure Align

Reasons Projects Fail	Agile Points of Failure
Inadequate Business Case; Undefined Objectives and Goals	Lack of Overall Product Design, Roadmap
Inadequate or Vague Requirements, Scope Creep	Adding Stories to an In Progress Iteration
Lack of a Solid Project Plan	Dependencies Not Identified, Cross-Team Priorities Not Coordinated
Unrealistic Timeframes and Budget; Inaccurate/Erroneous Estimates	Attempting to Take on Too Much in an Release, Iteration, or Sprint
Failure to track progress, manage risk & variances	High work in Progress, Allowing Technical Debt to Build Up
Poor Testing	Lack of Test Automation
Poorly Defined Roles & Responsibilities, Inadequate Resources	Teams Are Not Focused, Scrum Master is a Contributor
Weak project management	Product Owner Role Isn't Properly Filled
Ineffective or No Sponsorship	Lack of Sponsor Support
Poor or Inflexible Processes/Documentation	Inconsistent Processes

Take-Aways

- Define your engineering and management processes to ensure efficient operations to keep teams in synch and make metrics meaningful.
- Build a Capability Breakdown Structure that supports the vertical slices of Agile development to a level that allows technical dependencies between Epics and Teams to be identified with precedence logic in an IMS
- Changes and Trade-offs are a normal part of Development, but without an agreed plan and system definition, you won't be able to assess the full impact of the change (programmatically or technically)

References and Resources

- Scaled Agile Framework (www.scaledagileframework.com)
- NDIA Integrated Performance Management Division
<https://www.ndia.org/divisions/ipmd/division-guides-and-resources>
- Department of Defense Integrated Program Management Guidance:
(<https://www.acq.osd.mil/asda/ae/ada/ipm/policy-guidance.html#guides-references>)
- GAO Agile Assessment Guide
(<https://www.gao.gov/products/gao-20-590g>)
- CMMI (<https://cmmiinstitute.com/cmmi>)

Robin Pulverenti

Robin.Pulverenti@AdeptForceGroup.com

www.linkedin.com/in/robinpulverenti/

315-572-1415

